

[속제 3] ARM 명령어

다음은 2장에 배운 ARM 명령어이다. 각 명령어의 사용법을 이해하기 위해 아래 표 세 번째 컬럼(예)에 있는 명령어가 각각 수행되었을 때 수행결과를 확인한다.

ARM 어셈블리 언어

종류	명령어	예	의미	비고
산술	add	ADD r1, r2, r3	$r1 = r2 + r3$	레지스터 피연산자 3개
	subtract	SUB r1, r2, r3	$r1 = r2 - r3$	레지스터 피연산자 3개
데이터 전송	load register	LDR r1, [r2, #20]	$r1 = \text{Memory}[r2 + 20]$	워드를 메모리에서 레지스터로
	store register	STR r1, [r2, #20]	$\text{Memory}[r2 + 20] = r1$	워드를 레지스터에서 메모리로
	load register halfword	LDRH r1, [r2, #20]	$r1 = \text{Memory}[r2 + 20]$	하프워드를 메모리에서 레지스터로
	load register halfword signed	LDRSH r1, [r2, #20]	$r1 = \text{Memory}[r2 + 20]$	하프워드를 메모리에서 레지스터로
	store register halfword	STRH r1, [r2, #20]	$\text{Memory}[r2 + 20] = r1$	하프워드를 레지스터에서 메모리로
	load register byte	LDRB r1, [r2, #20]	$r1 = \text{Memory}[r2 + 20]$	바이트를 메모리에서 레지스터로
	load register byte signed	LDRSB r1, [r2, #20]	$r1 = \text{Memory}[r2 + 20]$	바이트를 메모리에서 레지스터로
	store register byte	STRB r1, [r2, #20]	$\text{Memory}[r2 + 20] = r1$	바이트를 레지스터에서 메모리로
	swap	SWP r1, [r2, #20]	$r1 = \text{Memory}[r2 + 20]$, $\text{Memory}[r2 + 20] = r1$	레지스터와 메모리 간의 원자적 교환
	mov	MOV r1, r2	$r1 = r2$	값을 레지스터로 복사
논리	and	AND r1, r2, r3	$r1 = r2 \& r3$	레지스터 피연산자 3개 비트 대 비트 AND
	or	ORR r1, r2, r3	$r1 = r2 r3$	레지스터 피연산자 3개 비트 대 비트 OR
	not	MVN r1, r2	$r1 = \sim r2$	레지스터 피연산자 3개 비트 대 비트 NOT
	logical shift left (optional operation)	LSL r1, r2, #10	$r1 = r2 \ll 10$	상수만큼 좌측 자리이동
	logical shift right (optional operation)	LSR r1, r2, #10	$r1 = r2 \gg 10$	상수만큼 우측 자리이동
조건부 분기	compare	CMP r1, r2	$\text{cond. flag} = r1 - r2$	조건부 분기를 위한 비교
	branch on EQ, NE, LT, LE, GT, GE, LO, LS, HI, HS, VS, VC, MI, PL	B EQ 25	if $(r1 == r2)$ go to $\text{PC} + 8 + 100$	조건 테스트; PC-상대 주소
무조건 분기	branch (always)	B 2500	go to $\text{PC} + 8 + 10000$	분기
	branch and link	BL 2500	$r14 = \text{PC} + 4$; go to $\text{PC} + 8 + 10000$	프로시저 호출용

1) 산술 명령어 그룹에 대한 부분이다. 초기 값이 $r2=0x76543210$ (16진수), $r3=0x98765432$, 상태 bit들 $N=0$, $Z=0$, $C=1$, $V=0$ 라고 하자. 산술명령어 ADDS $r1, r2, r3$ 수행된 후 **r1** 값을 16진수로 표기하시오. 또한 상태 bit N , Z , C , V 값이 수정되었다면 그 값을 쓰시오. ADDS 대신 SUBS가 수행된 후(초기 조건은 위와 동일) **r1** 값 및 상태비트 값들을 쓰시오. SUB 명령어는 $A - B = A + (B \text{의 } 2 \text{의 보수})$ 방식으로 계산한다고 하자.

2) 데이터 전송 명령어 그룹에 대한 부분이다. 초기값은 $r1=1000$ (10진수),

r2=2000(10진수), 메모리 [2000(10진수)]에는 00, [2001]에는 01, [2002]에는 02,... [2020]에는 20 (10진수)이 저장되어 있다고 가정한다. 위 표에서 LDR r1, [r2, #20] 명령어부터 MOV r1, r2 까지 각각 독립적으로 수행(각 명령어 입장에서 초기 조건은 동일)한다고 하고 수행후 레지스터 및 메모리 내용 변경내용을 각각 설명하시오. CPU는 Little Endian 32bit 방식으로 데이터를 저장한다고 가정한다.

3) 논리 명령어 그룹에 대한 부분이다. 초기 값이 r2=0x76543210, r3=0x98765432, 상태 bit들 N=0, Z=0, C=1, V=0 라고 하자. 첫 두 개의 명령어는 AND, ORR 대신 ANDS, ORRS가 수행된다고 하고 나머지 명령어들은 표에 있는 대로 수행된다고 하자. 수행후 r1 값, 상태비트가 수정되었다면 그 값을 각각 쓰시오. 논리동작 명령어 뒤의 접미사 S에서는 V는 그대로 두고 나머지 N, Z, C 만 update한다.

4) 조건부 분기 명령어 그룹에 대한 부분이다. ①초기 값이 r1=0x76543210, r2=0x98765432, PC=1000 (10진수), 상태 bit들 N=0, Z=0, C=1, V=0 라고 하자. CMP수행후 변경된 상태 비트값을 쓰시오 (CMP는 접미사 S가 없지만 CMPS의 의미로 사용된다). CMP 수행후 PC=1004 (10진수)로 변경되고 변경된 상태 비트가 유지된다. BEQ 25 수행후 PC값을 쓰시오 ② 초기 값이 r1=0x87654321, r2=0x87654321, PC=1000 (10진수), 상태 bit들 N=1, Z=0, C=1, V=0 라고 하자. CMP 수행후 변경된 상태 비트값을 쓰시오. CMP 수행후 PC=1004로 변경되고 변경된 상태 비트가 유지된다. BEQ 25 수행후 PC값을 쓰시오.

5) 무조건부 분기 명령어 그룹에 대한 부분이다. ①초기 값이 PC=1000 (10진수), B 200 (10진수) 수행후 PC 값을 쓰시오. ② 초기 값이 PC=1000 (10진수), BL 200 (10진수) 수행후 PC 및 LR 값을 쓰시오.